





Sequences Modeling with Deep Learning

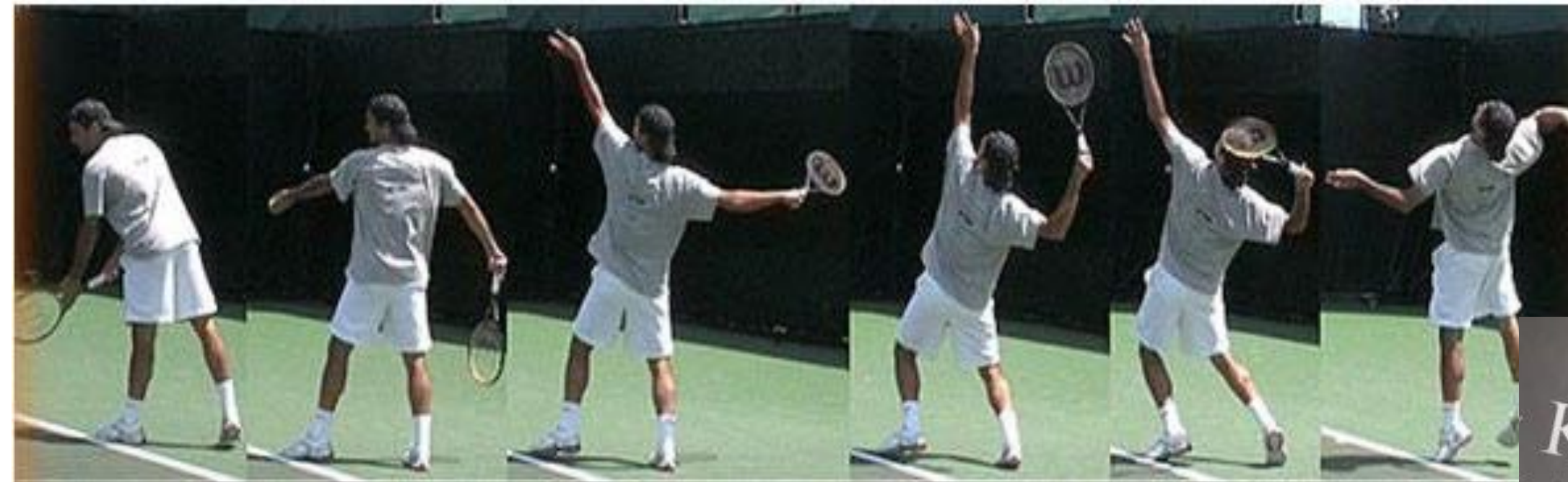
Marc'Aurelio Ranzato

Facebook AI Research

ranzato@fb.com

NYU - DS-GA 1008, 20 March 2017

Data is often sequential in nature



The image shows a close-up of an ASX (Australian Securities Exchange) electronic stock market board. The board displays a list of stocks with their respective bid, offer, last price, and volume. The ASX logo is visible in the top left corner.

STOCK	BID	OFFER	LAST	VOL	STOCK	BID	OFFER
EUR GROUP	0.060	0.070	0.000	0	FARM PRIDE	0.100	0.140
EUROGOLD	0.098	0.140	0.000	0	FE LIMITED	0.026	0.030
EUROP GAS	0.325	0.335	0.335	77T	FEO.AX	0.120	0.130
EUROZ	1.000	1.020	1.000	4T	FERROWEST	0.024	0.033
EVOLUTION	1.935	1.940	1.935	2M	FERRUM	0.052	0.057
EVZ LTD	0.041	0.050	0.050	5T	FIDUCIAN	0.800	0.810
EXALT RES	0.000	0.000	0.000	0	FIE.AX	0.110	0.125
EXC.AX	0.040	0.049	0.040	50T	FINBAR	1.075	1.080
EXCALIBUR	0.001	0.002	0.000	0	FINDERS	0.200	0.220
EXCELA	0.010	0.090	0.000	0	FIRESTONE	0.008	0.009
EXCELSIOR	0.190	0.195	0.190	30T	FIRSTFOLIO	0.014	0.015
EXCO RES	0.260	0.265	0.260	5HT	FISSION EN	0.020	0.035
EXOMA ENER	0.072	0.075	0.072	35T	FITZROYRES		
EZAAX	0.430	0.480	0.430				

Questions

— Deep learning tools to learn from and to predict sequences

- can standard tools like CNNs suffice?
- how about RNNs?

— fundamental problems when dealing with sequences

- is the sequential structure important for the prediction task?
- how to leverage structure at the input?
- how to deal with large output spaces? how to predict and what loss function to use?
- how to deal with variable length inputs/outputs? how to align sequences?

TL;DR...

There is no general rule of thumb, it depends on the task and constraints at hand.

Next, we will review several examples.

Learning Scenarios

		Output Sequential?	
		no	yes
Input Sequential?	no		?
	yes	?	?

Learning Scenarios: sequence -> single label

		Output Sequential?	
		no	yes
Input Sequential?	no		?
	yes	X	?

Examples:

- text classification
- language modeling
- action recognition
- music genre classification

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

“I’ve had this place bookmarked for such a long time and I finally got to go!! I was not disappointed...

“ -> **positive rating**

Text classification

“Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on” -> **science**

General problem:

Given a document (ordered sequence of words), predict a single label.

Challenge:

Efficiency VS accuracy trade-off.

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

“I’ve had this place bookmarked for such a long time and I finally got to go!! I was not disappointed...

“ -> **positive rating**

Text classification

“Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on” -> **science**

Approach:

Embed words in \mathbb{R}^d -> average embeddings -> apply a linear classifier.

Word order is lost. This partially remedied by embedding n-grams.

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

“I’ve had this place bookmarked for such a long time and I finally got to go!! I was not disappointed...”

“ -> **positive rating**

negative



Text classification

“Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on” -> **science**

Approach:

Embed words in \mathbb{R}^d -> average embeddings -> apply a linear classifier.

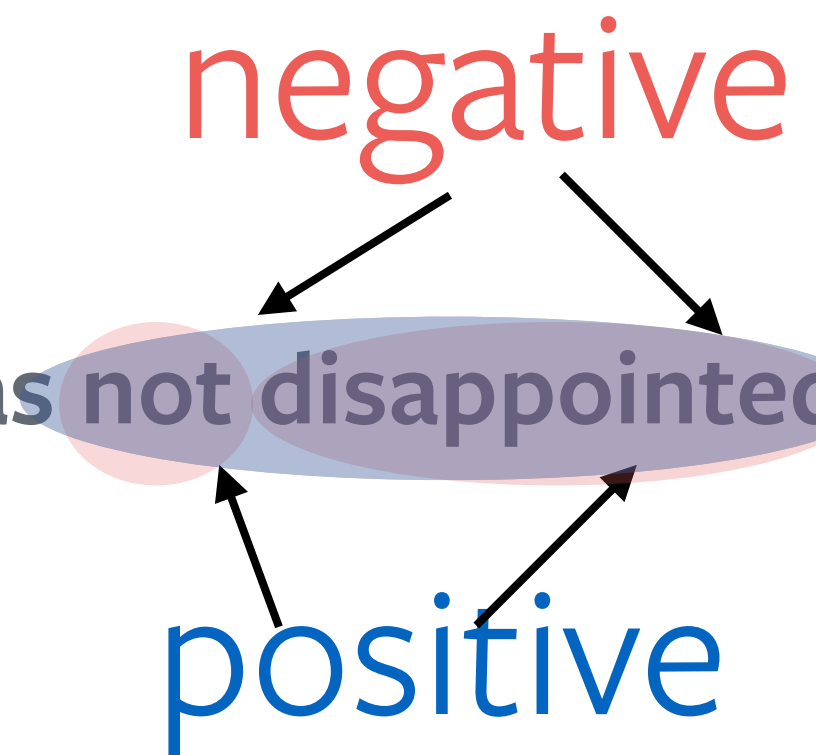
Word order is lost. This partially remedied by embedding n-grams.

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

“I’ve had this place bookmarked for such a long time and I finally got to go!! I was not disappointed...”
“ -> **positive rating**



Text classification

“Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on” -> **science**

Approach:

Embed words in \mathbb{R}^d -> average embeddings -> apply a linear classifier.
Word order is lost. This partially remedied by embedding n-grams.

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

“I’ve had this place bookmarked for such a long time and I finally got to go!! I was not disappointed...”

“ -> **positive rating**

negative

positive

Text classification

“Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on” -> **science**

Conclusion:

In this application (so far), bagging n-grams (n=1, 2, ...) works the best and is very efficient. No need to deal with sequential nature of the input!

Sequence->Single Label: Language Modeling

Example

“Neural networks or connectionist systems are a computational ???” Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

$$p(w_t | w_{t-1} \dots w_1)$$

Challenges:

- very large vocabularies (> 100,000 words)
- long range dependencies (overall if working at the character level)

Sequence->Single Label: Language Modeling

Example

“Neural networks or connectionist systems are a computational ???” Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

$$p(w_t | w_{t-1} \dots w_1)$$

Approaches:

- n-grams
- RNNs Exploring the limits of language modeling, Jozefowicz et al. 2016
- CNNs (more recently) Language modeling with gated convolutional networks, Dauphin et al. 2016

Sequence->Single Label: Language Modeling

Example

“Neural networks or connectionist systems are a computational ???” Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

Approaches:

- **n-grams:** count-based, works well for head of distribution.

In order to estimate:

$$p(w_t | w_{t-1} \dots w_1)$$

we first make the Markov assumption that:

$$p(w_t | w_{t-1} \dots w_1) = p(w_t | w_{t-1} \dots w_{t-n+1})$$

and then we simply count:

$$p(w_t | w_{t-1} \dots w_{t-n+1}) = \frac{\text{count}(w_{t-n+1} \dots w_t)}{\text{count}(w_{t-n+1} \dots w_{t-1})}$$

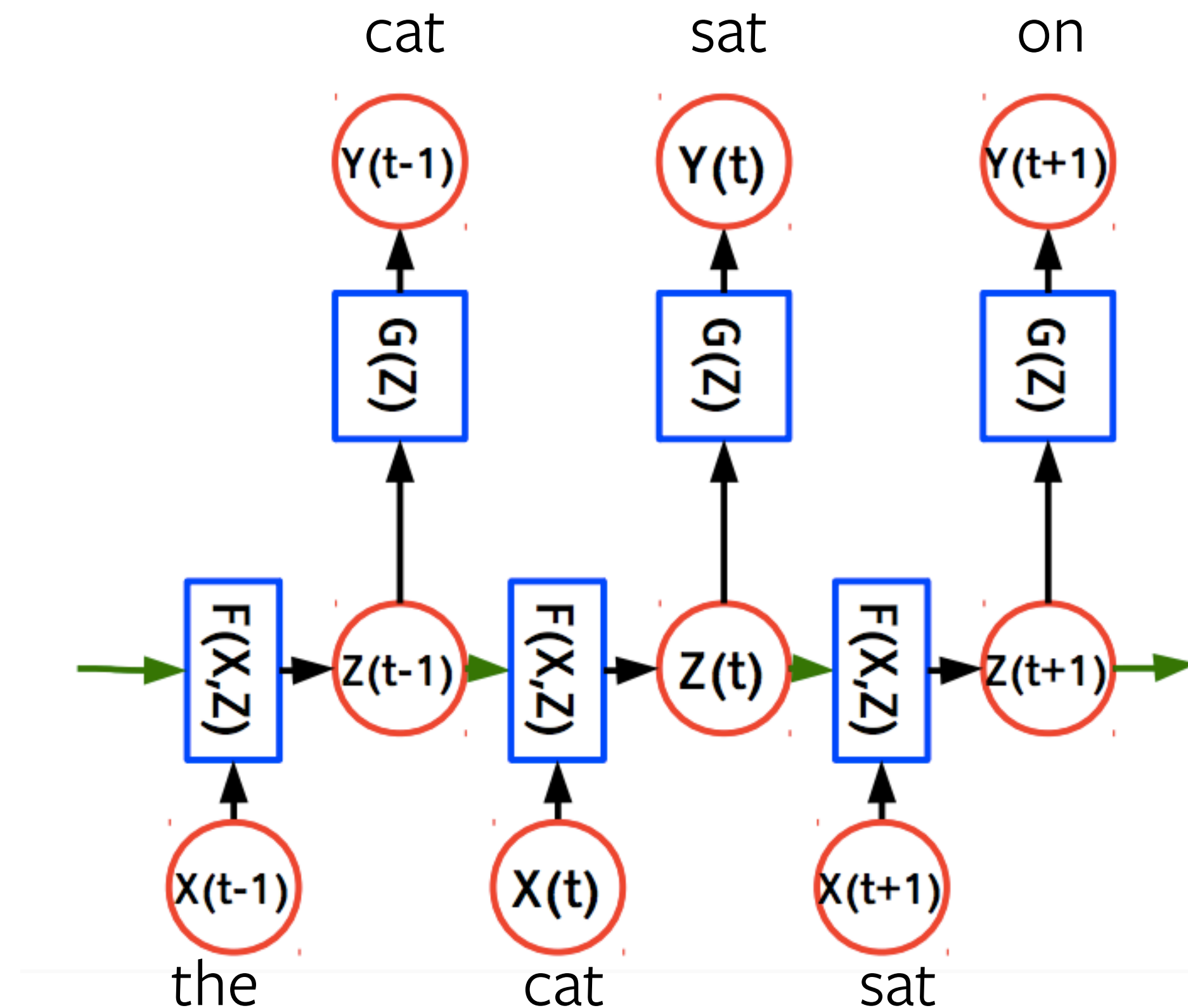
Sequence->Single Label: Language Modeling

Example

“Neural networks or connectionist systems are a computational ???” Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

Approaches:

- RNNs



Y. LeCun's diagram

Sequence->Single Label: Language Modeling

Example

“Neural networks or connectionist systems are a computational ???” Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

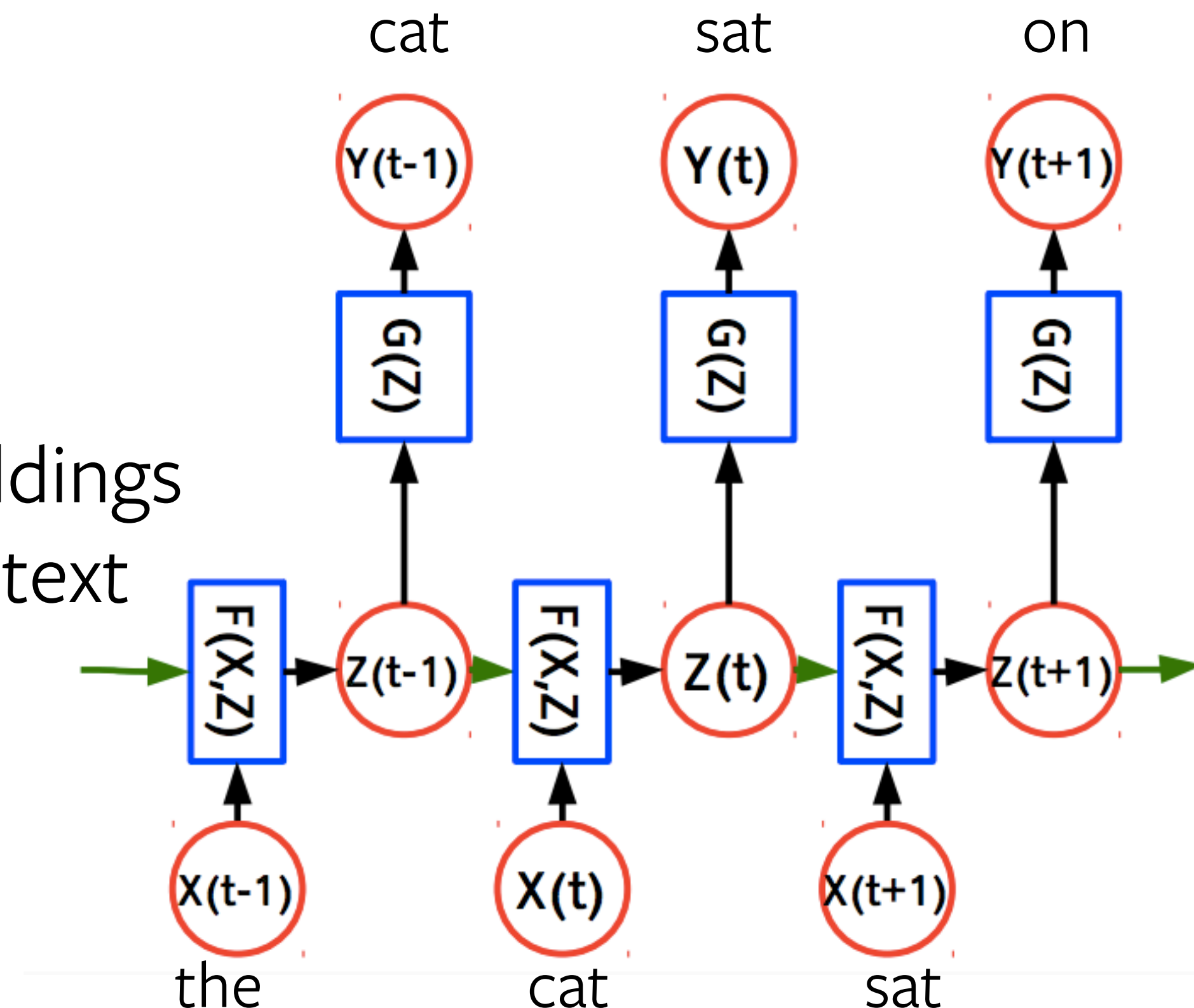
Approaches:

- RNNs

- + it generalizes better thanks to embeddings
- + it can more easily capture longer context
- it's sequential, tricky to train

Fun demo with a charRNN:

<http://www.cs.toronto.edu/~ilya/rnn.html>



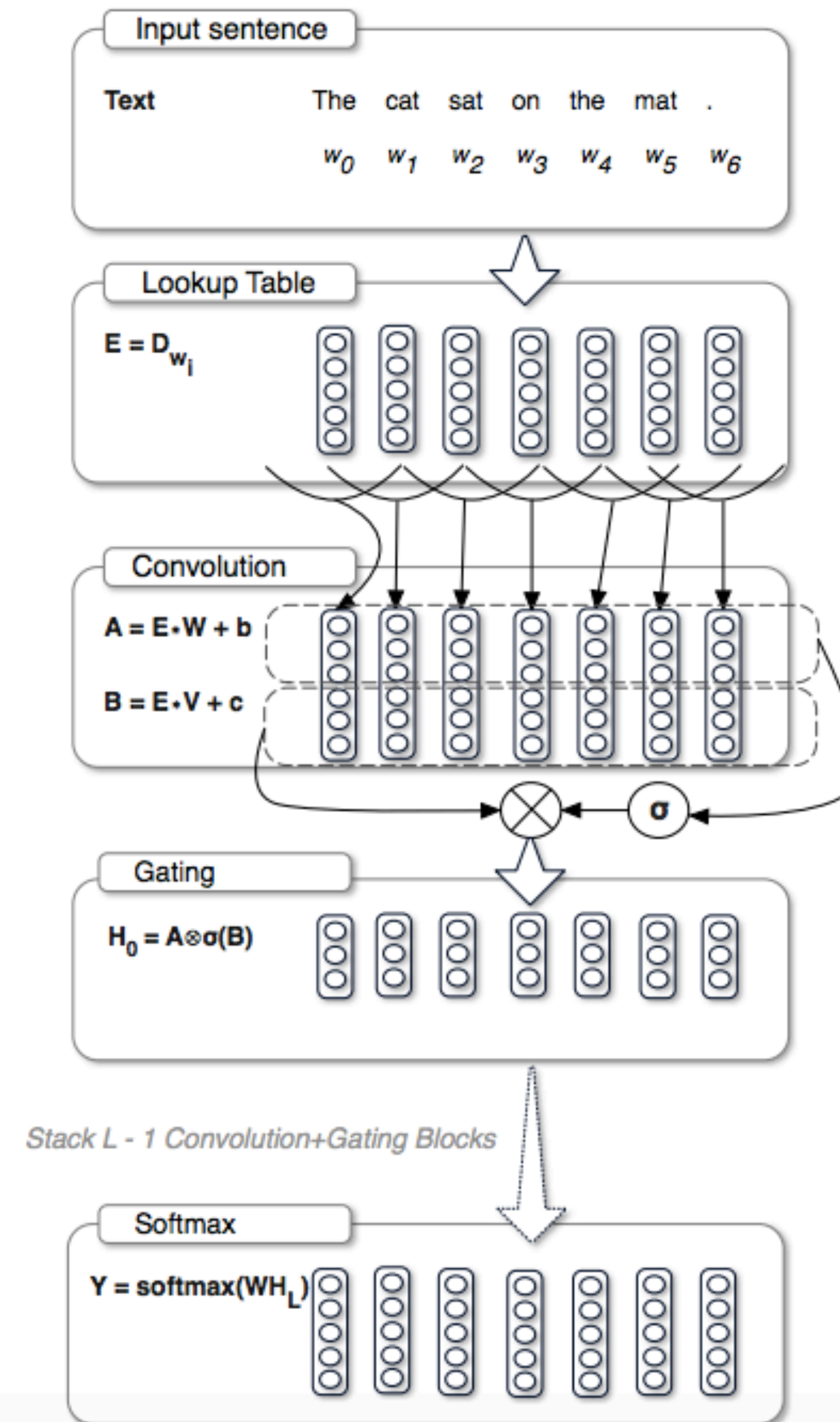
Y. LeCun's diagram

Sequence->Single Label: Language Modeling

Approaches:

- CNNs

- + same generalization as RNN
- + more parallelizable than RNNs
- fixed context



Language modeling with gated convolutional networks,
Dauphin et al. 2016

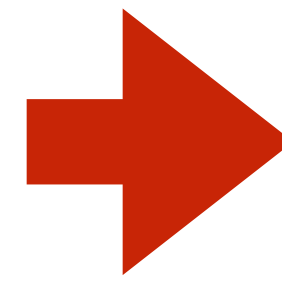
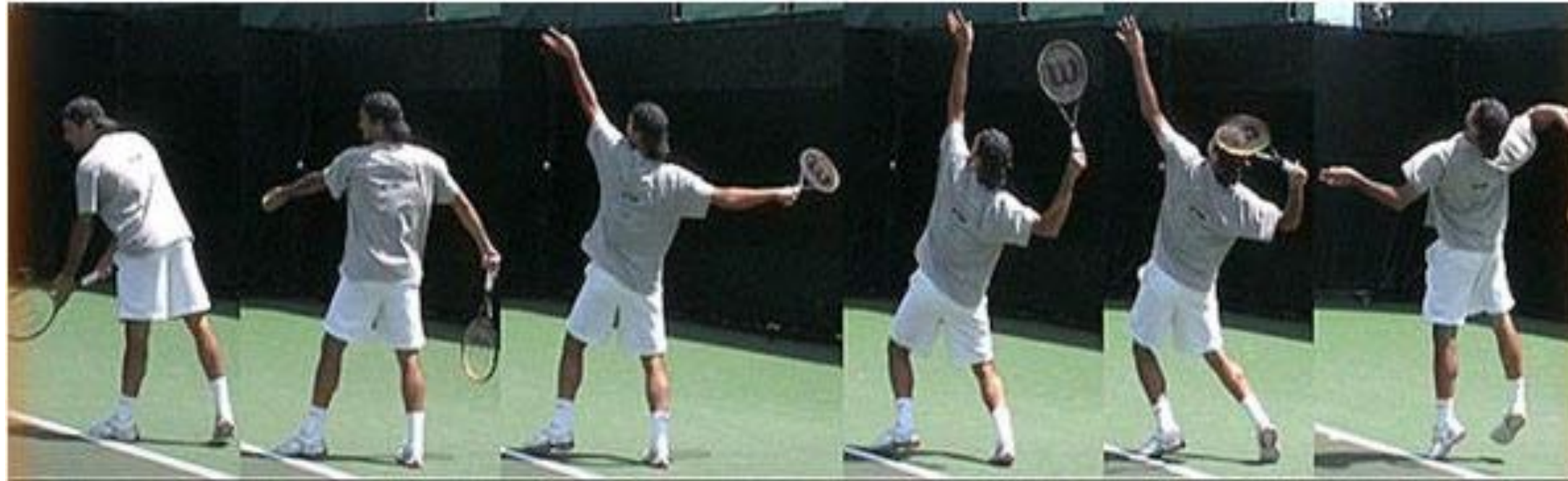
Sequence->Single Label: Language Modeling

Conclusion:

In language modeling, it is essential to take into account the sequential structure of the input.

RNNs/CNNs work the best at the moment.

Sequence->Single Label: Action Recognition

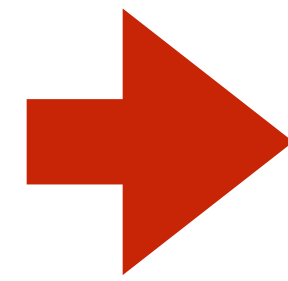
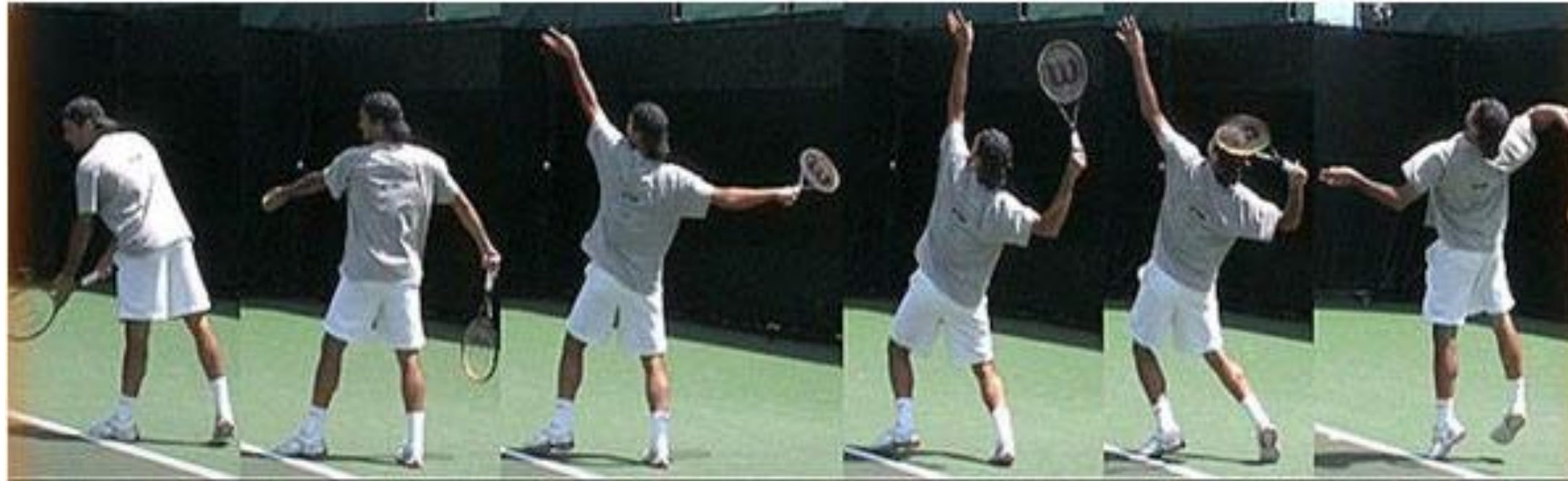


Playing Tennis

Challenges:

- how to aggregate information over time
- computational efficiency

Sequence->Single Label: Action Recognition



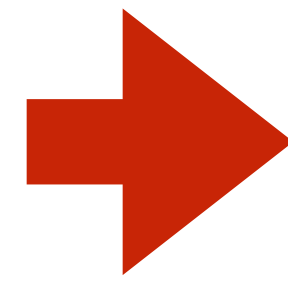
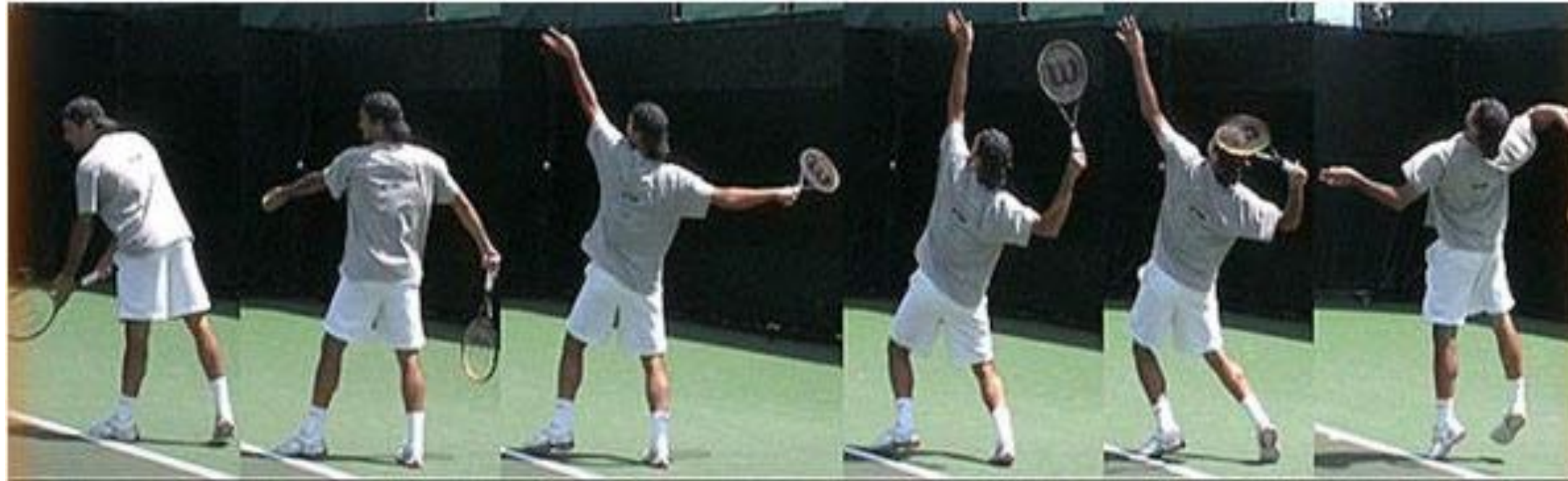
Playing Tennis

Approaches:

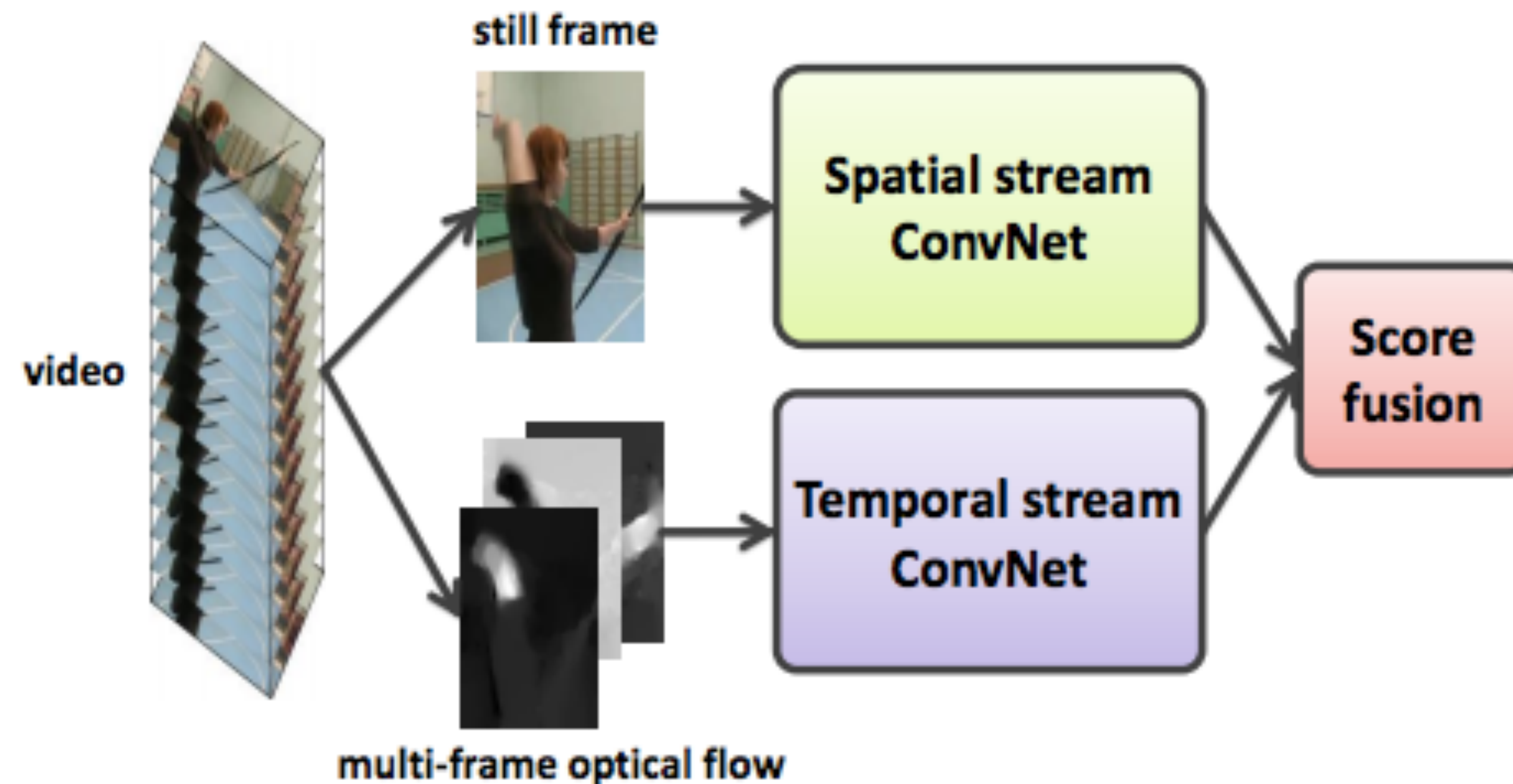
- CNN on static frames -> feature pooling over time -> classification. Possibly augmented with optical flow or (learned) temporal features.

Current large datasets have peculiar biases. E.g.,: one can often easily recognize the action from static frames by just looking at the context....

Sequence->Single Label: Action Recognition

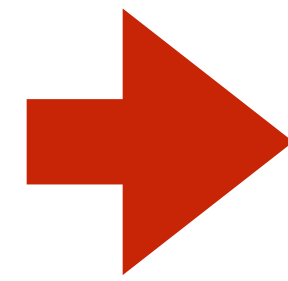
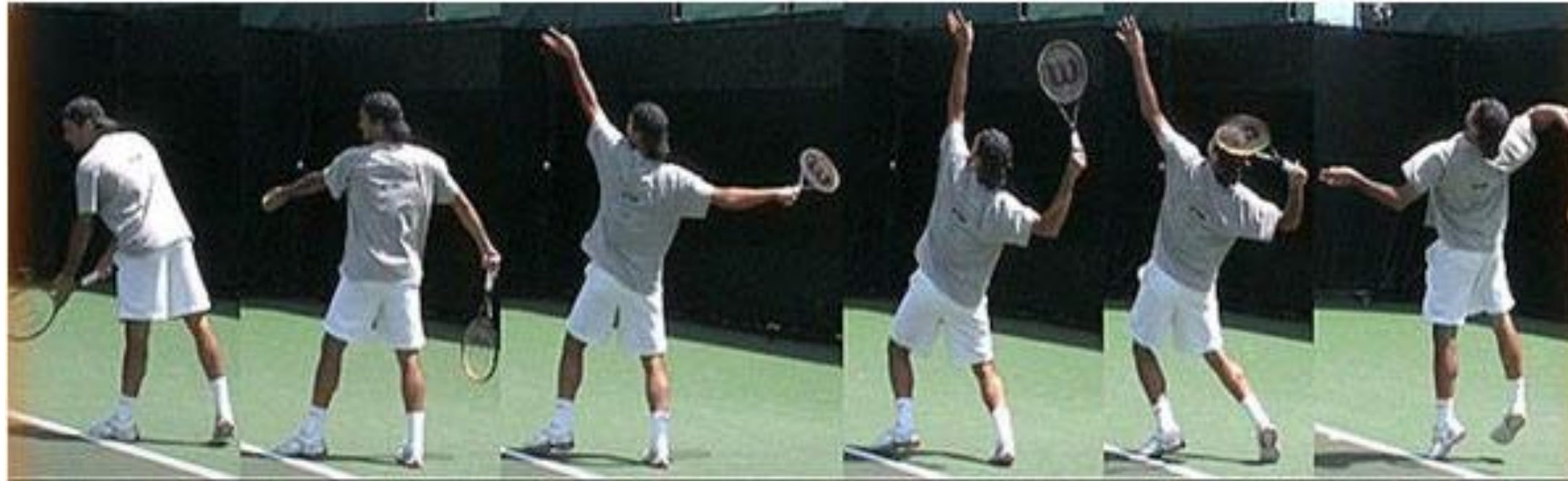


Playing Tennis



Two stream convolutional network for action recognition in videos. Simonyan et al. NIPS 2014

Sequence->Single Label: Action Recognition



Playing Tennis

Conclusion:

Methods and approaches heavily depend on the dataset used. Sometimes, the sequential structure does not add much information, if the label already correlates well with what can be found in static frames.

Learning Scenarios: single input -> sequence

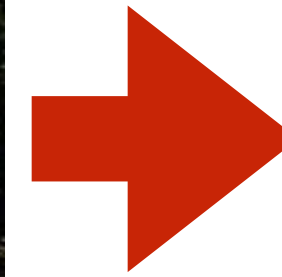
		Output Sequential?	
		no	yes
Input Sequential?	no		X
	yes	X	?

Example:

- image captioning

Single input -> sequence: image captioning

Example:



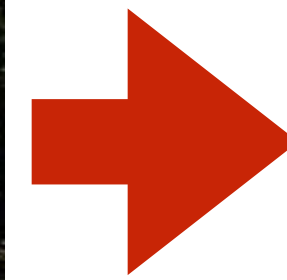
A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Challenge:

- how to deal with multiple modalities.
- what to look for and where to look in the input image.
- uncertainty in the output: there are many good captions for a given image.
- What is a good metric of success?

Single input -> sequence: image captioning

Example:



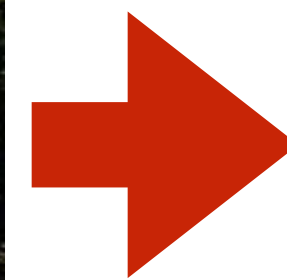
A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Approach:

Pre-train a CNN to extract features from the image, and generate text conditioning an RNN with the image features.

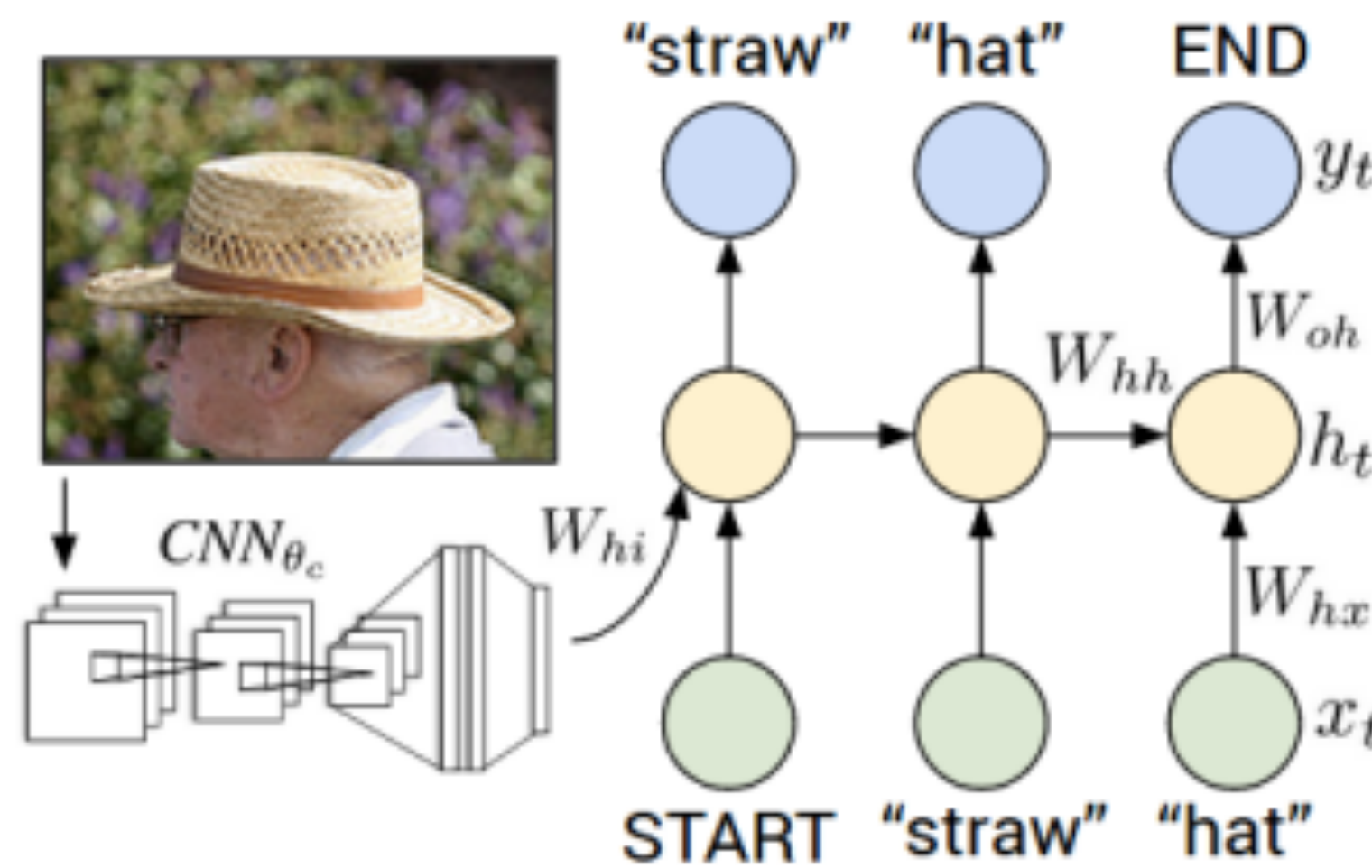
Single input -> sequence: image captioning

Example:



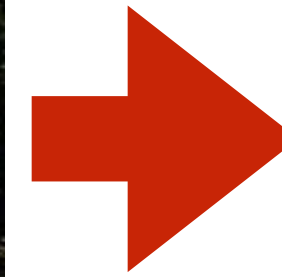
A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Approach:



Single input -> sequence: image captioning

Example:



A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Conclusion:

It is easy to condition a language model (RNN or CNN based) with additional context, and ultimately map a static object into a sequence.

This however heavily relies on good pre-trained (on large labeled datasets) image features.

Learning Scenarios: sequence -> sequence

Examples:

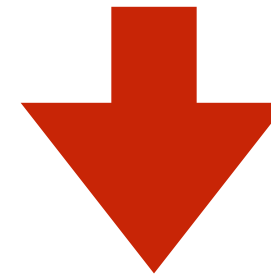
- machine translation
- summarization
- speech recognition
- OCR
- video frame prediction

		Output Sequential?	
		no	yes
Input Sequential?	no		X
	yes	X	X

Sequence -> Sequence: machine translation

Example:

ITA: Il gatto si e' seduto sul tappetino.



EN: The cat sat on the mat.

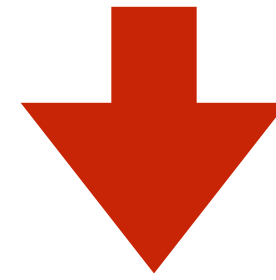
Challenges:

- alignment: input/output sequences may have different length
- uncertainty (1-to-many mapping: many possible ways to translate)
- metric: how to automatically assess whether two sentences mean the same thing?

Sequence -> Sequence: machine translation

Example:

ITA: Il gatto si e' seduto sul tappetino.

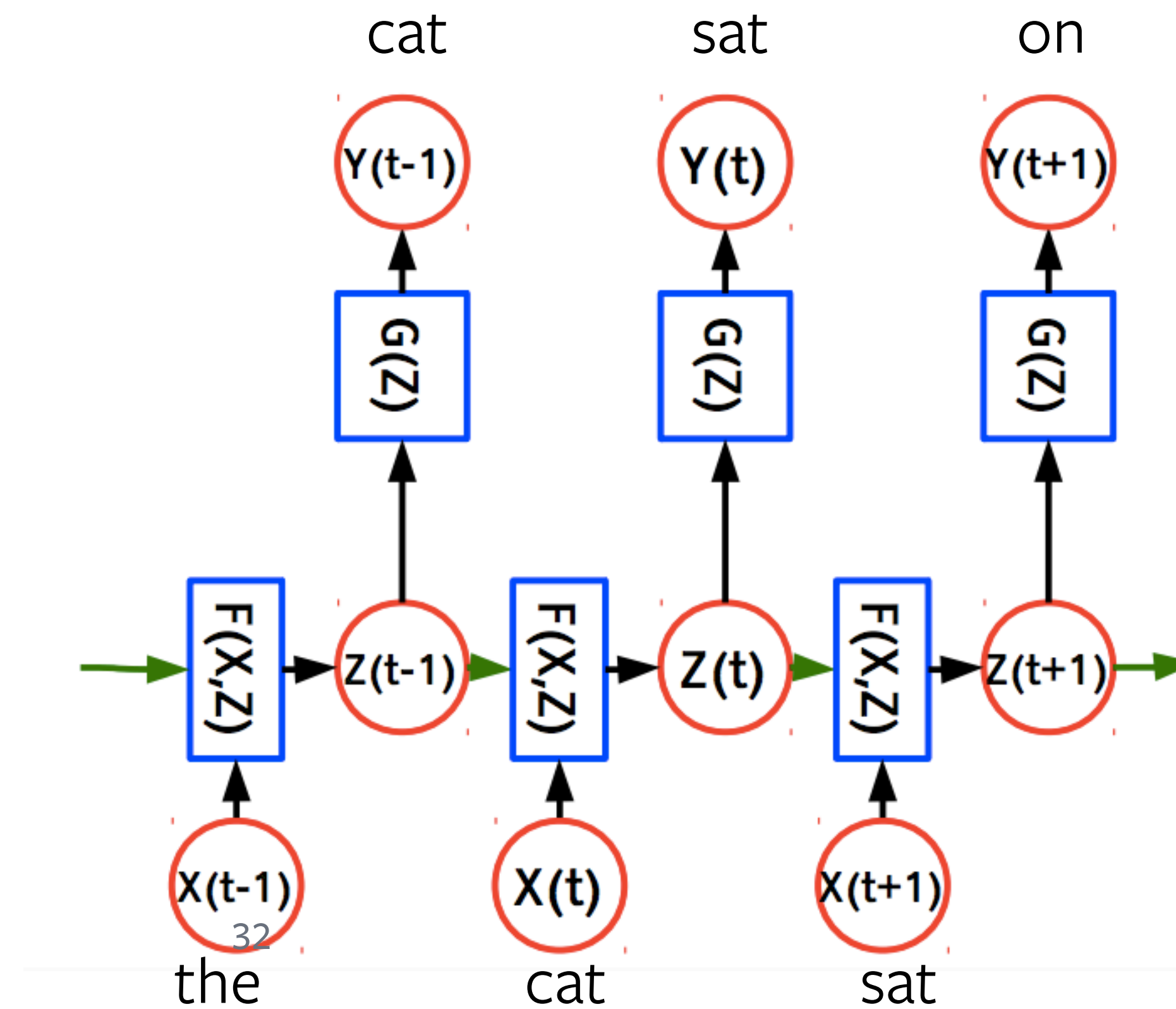


EN: The cat sat on the mat.

Approach:

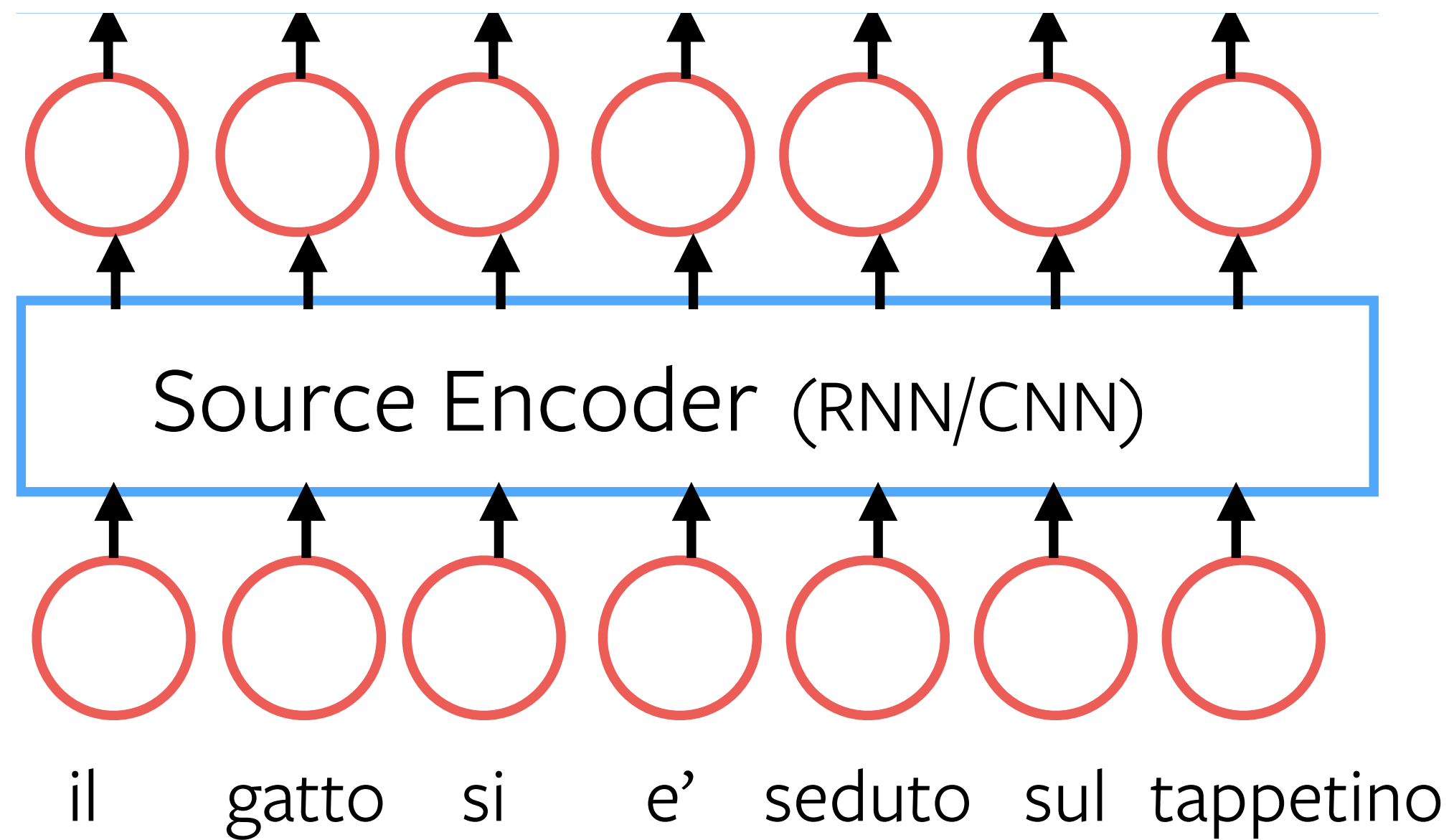
Have one RNN to encode the source sentence, and another RNN to predict the target sentence. The target RNN learns to (soft) align via attention.

Sequence -> Sequence: machine translation

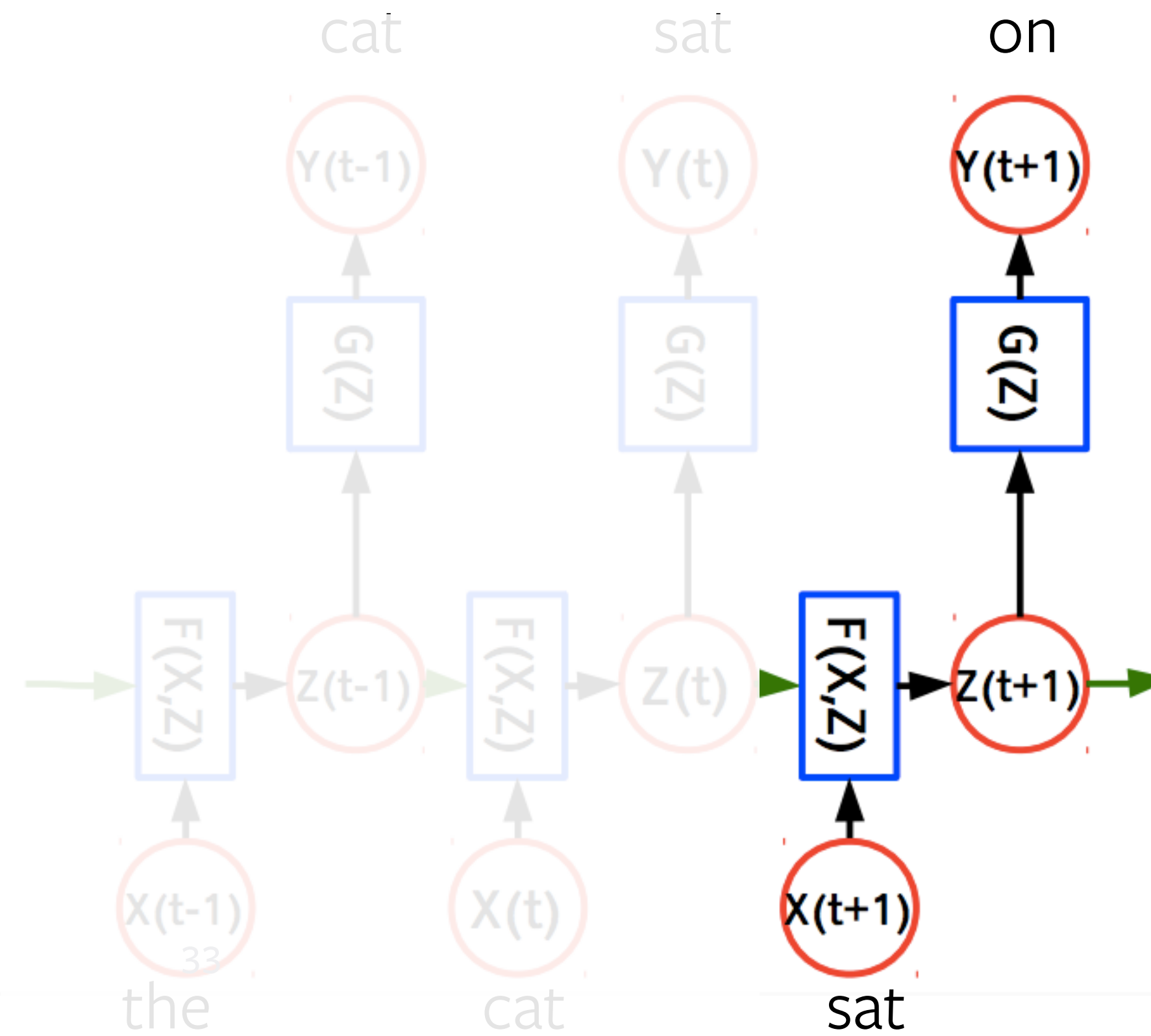


Source

1) Represent source



Target

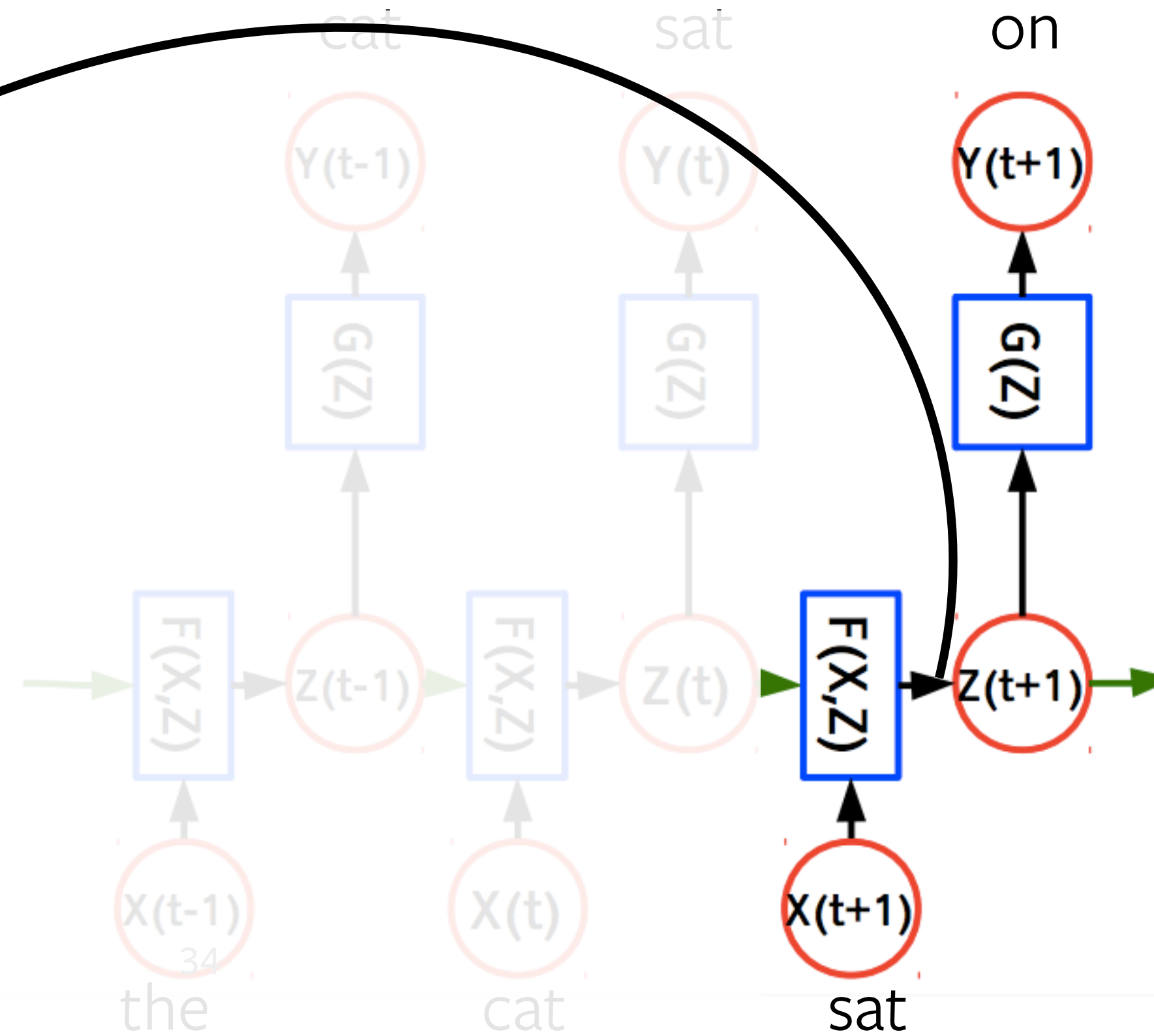
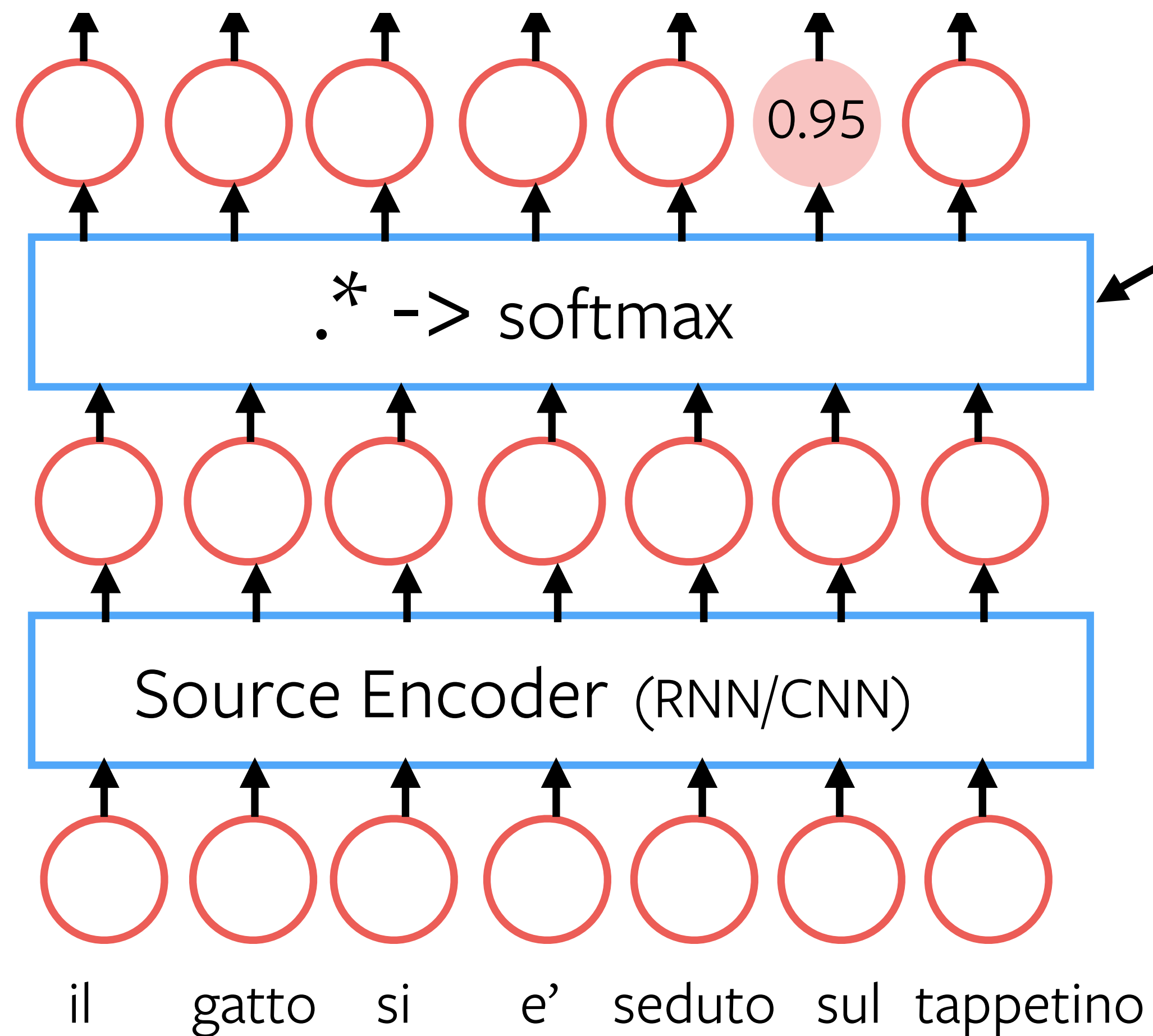


Y. LeCun's diagram++

Source

Target

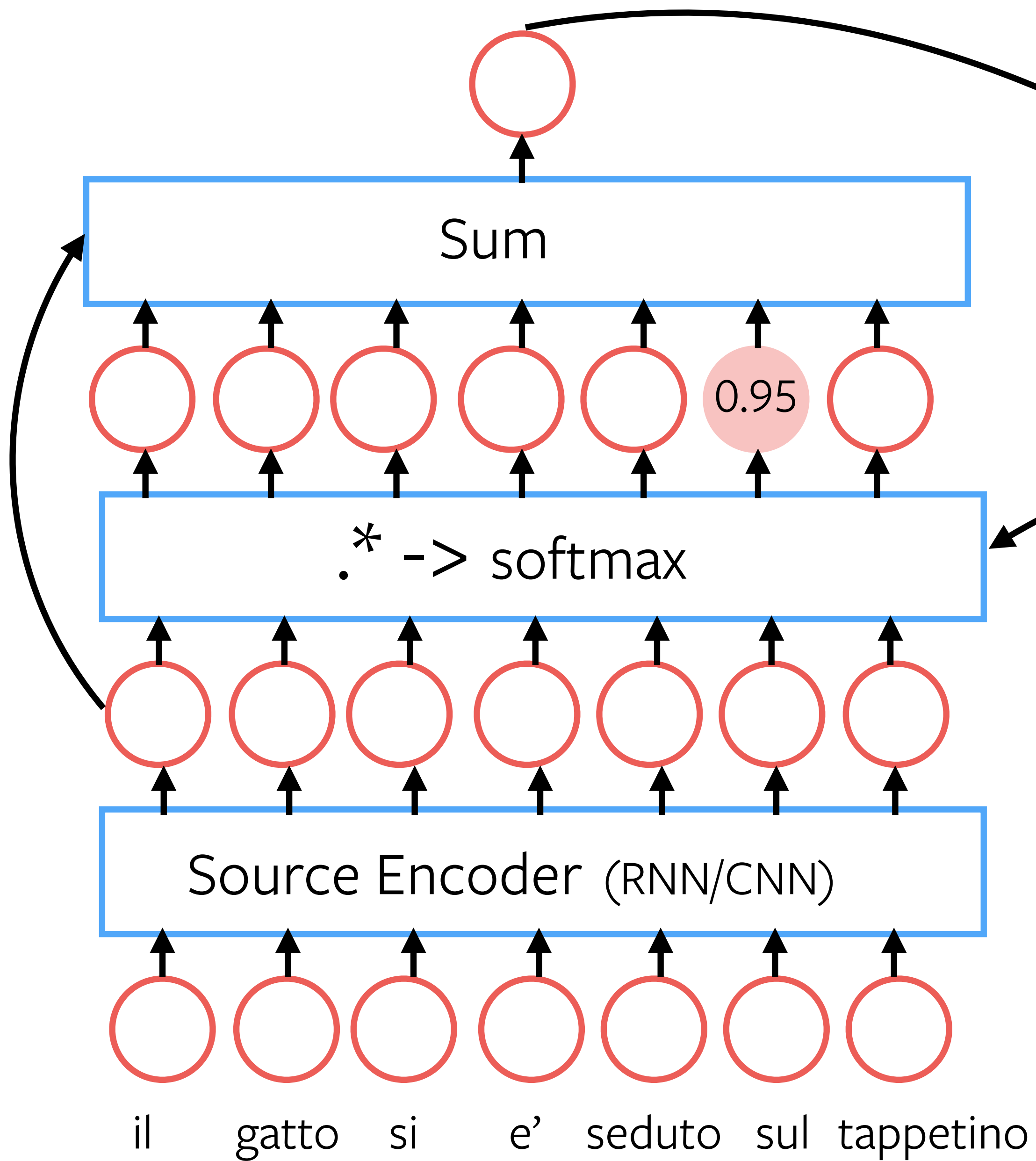
2) score each source word (attention)



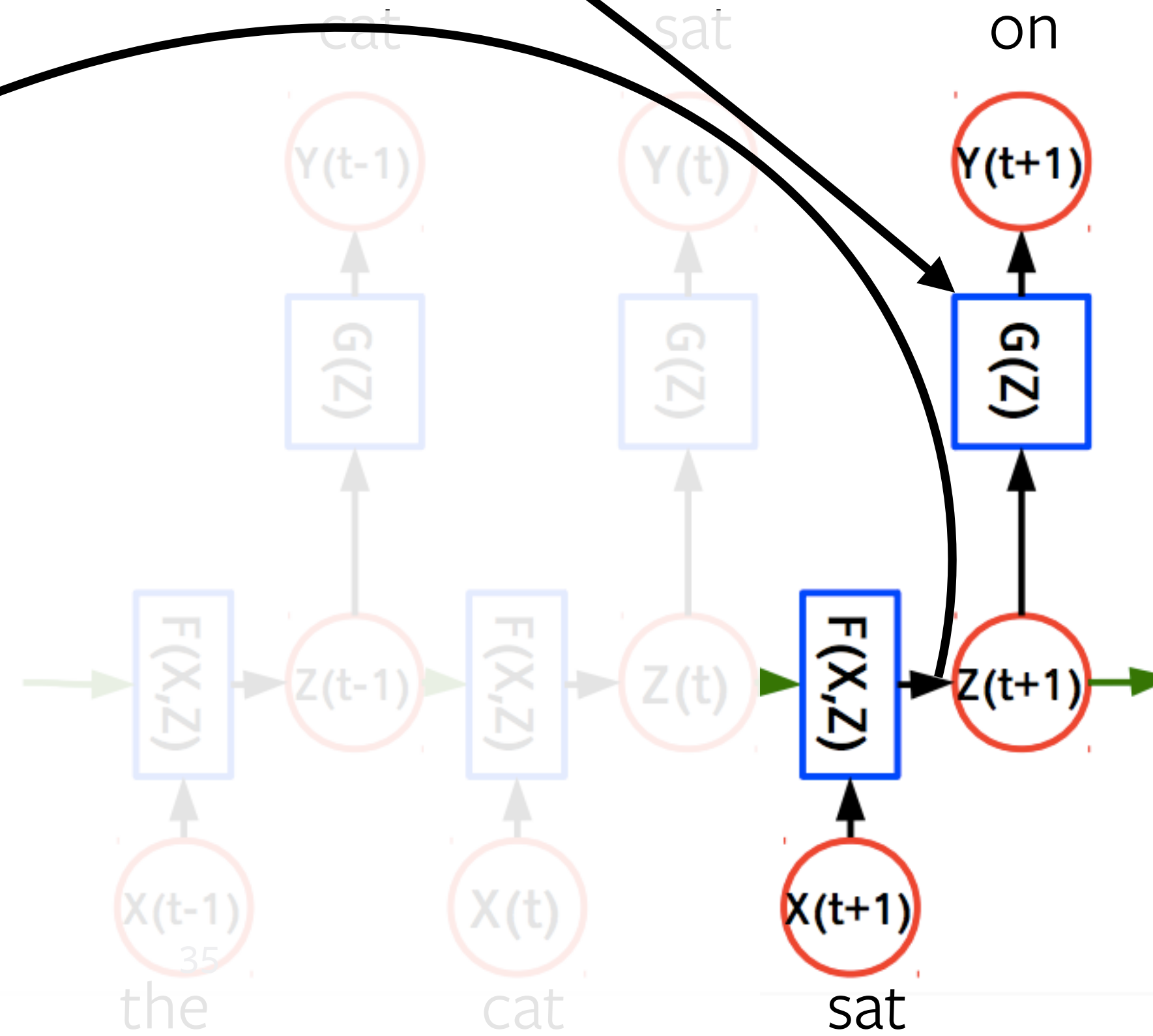
Y. LeCun's diagram++

Source

Target



3) combine target hidden with source vector

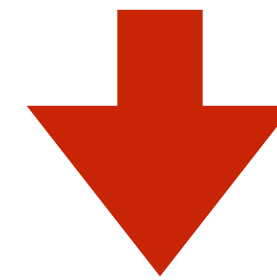


Y. LeCun's diagram++

Sequence -> Sequence: machine translation

Example:

ITA: Il gatto si e' seduto sul tappetino.



EN: The cat sat on the mat.

Notes:

- + source and target sentence can have any length, it works well on long sentences too!
- + it learns to align implicitly.
- + RNN can be replaced with CNNs. [A convolutional encoder model for NMT, Gehring et al. 2016](#)
- + it generates fluent sentences.
- It has trouble dealing with rare words, exact choice of words.
- It is typically trained like a language model (cross-entropy), good for scoring but not for generation.

Sequence -> Sequence: machine translation

Conclusions:

- + attention (gating) mechanism is rather general and it can be used for:
 - + dealing with variable length inputs, as it “softly select one”
 - + implicit alignment, which is discovered by the model as needed
 - + to perform rounds of “reasoning” (e.g., “hops” in memory networks)
- + the same mechanism has been used to image captioning, summarization, etc.
- word level loss function (cross entropy for predicting the next word) is sub-optimal for the generation task.

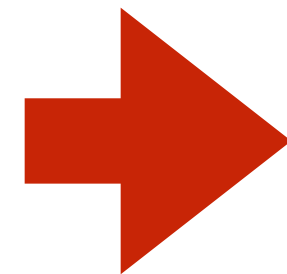
Sequence level training with RNNs, Ranzato et al. ICLR 2016

An actor-critic algorithm for sequence prediction, ICLR 2017

Sequence-to-sequence learning as beam-search optimization, EMNLP 2016

Sequence -> Sequence: OCR

Example 1



Sequence -> Sequence: OCR

Example 2



Sequence -> Sequence: OCR

Example 2

Thomas B Anderson
Mary B Anderson
PO BOX 678
2063 Main Street
Anywhere USA 12345-6789

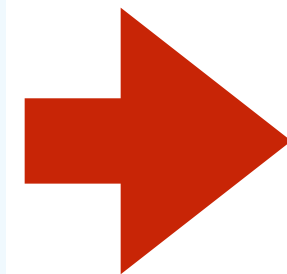
DATE 1/14/14

1001

PAY TO THE ORDER OF Sample Company \$ 200.00
two hundred + ⁰⁰/₁₀₀ dollars Dollar

MEMO MONTHLY BILL John Sample

⑆ 222370440 ⑆ 123456789123 ⑆ 1001



“200”

Sequence -> Sequence: OCR



➔ “200”

Challenges:

- digit segmentation is not observed; there can be several segmentations that are correct (i.e., yield correct transcription).
- variable length.
- design of loss function.
- very large number of valid output sequences.

Sequence -> Sequence: OCR



➔ “200”

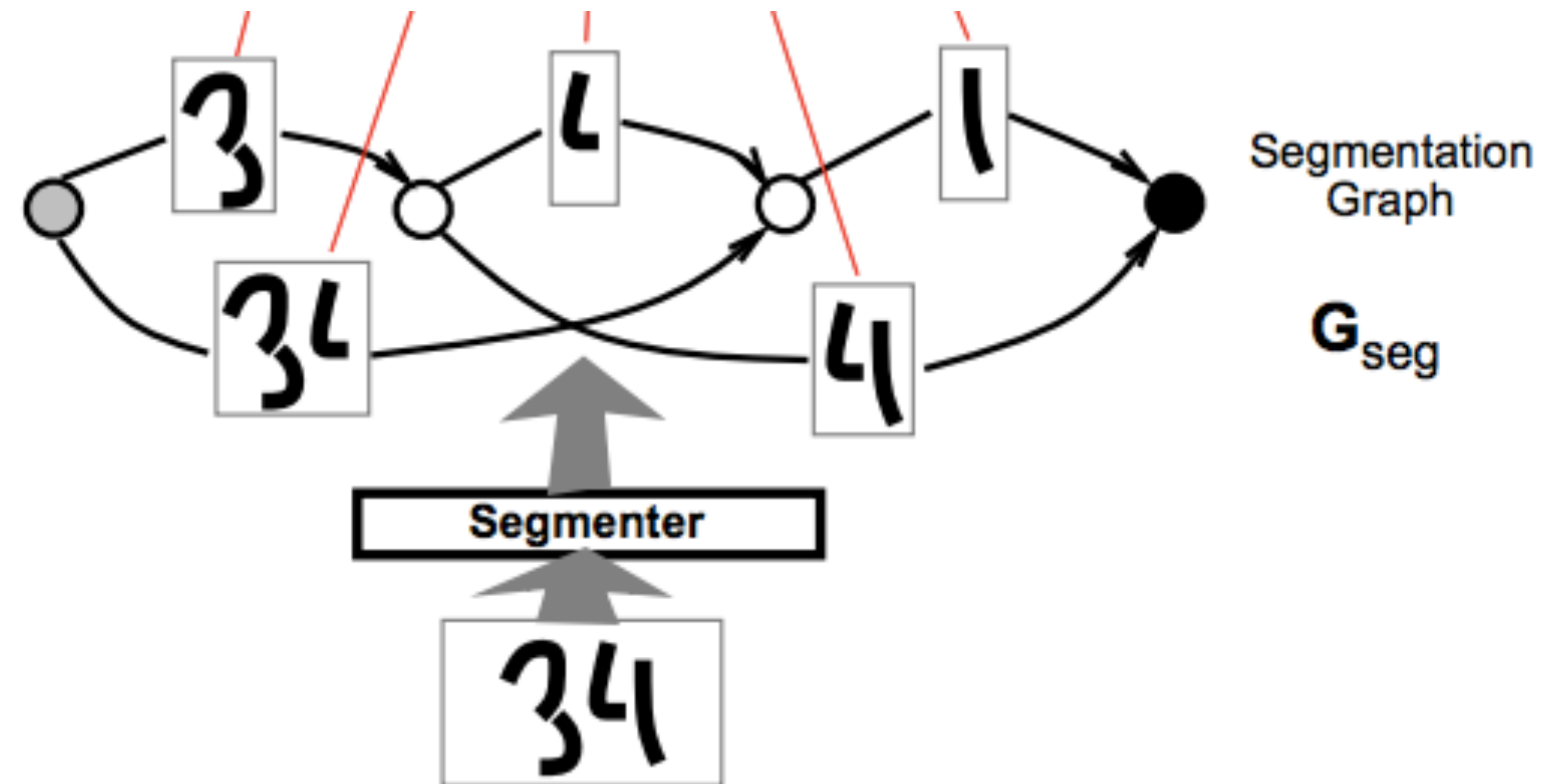
Approach:

- pre-train a CNN on single handwritten digits.
- over-segment and produce a lattice of possible “interpretations”.
- apply graph-transformer networks with a log-likelihood loss over sequences or margin loss.

Global training of document processing systems with graph transformer networks, Bottou et al. CVPR 1997
Gradient-based learning applied to document recognition, LeCun et al. IEEE 1998
Deep structured output learning for unconstrained text⁴² recognition, Jaderberg et al. ICLR 2015

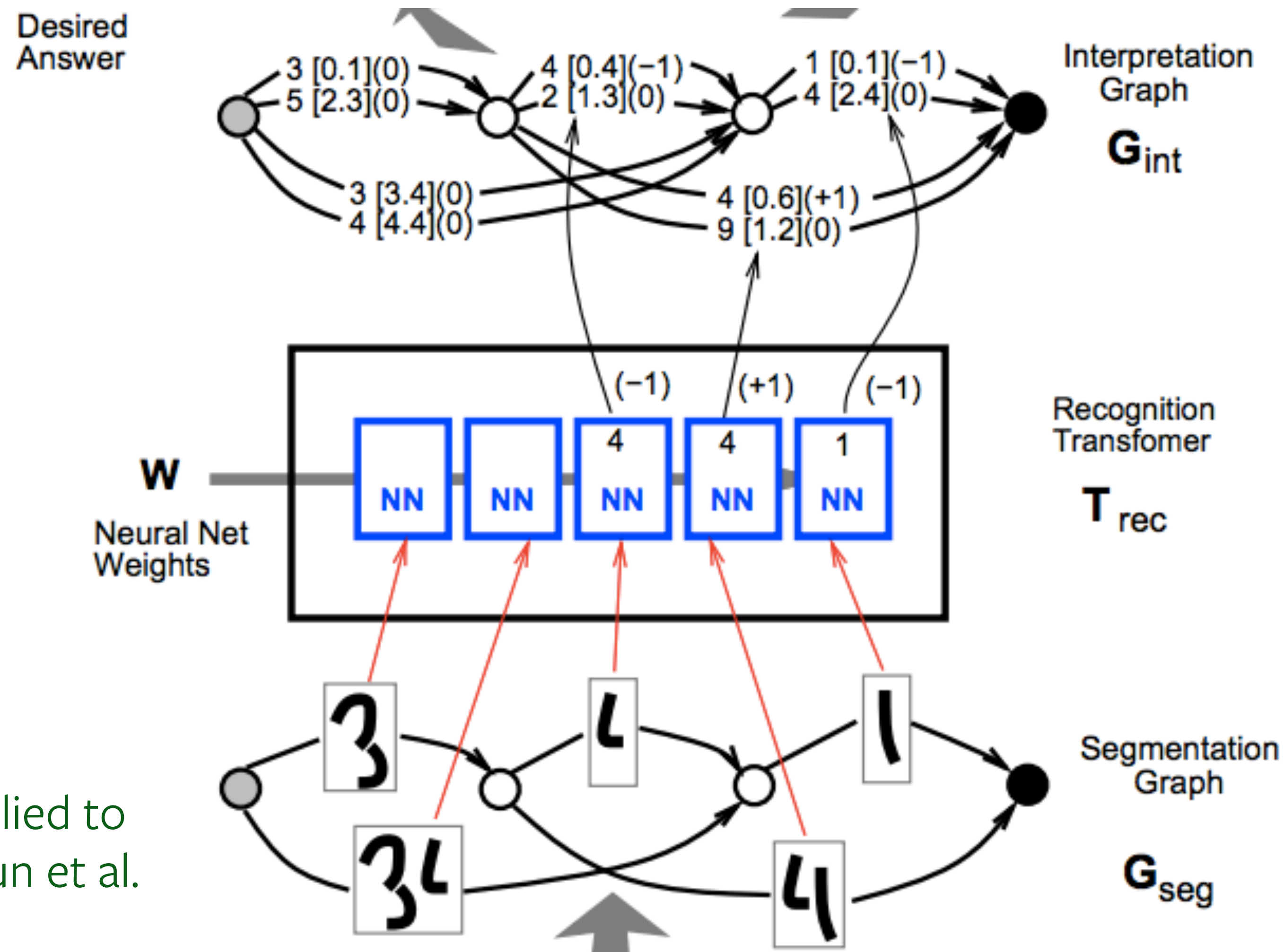
Sequence -> Sequence: OCR

Step1: over-segment & produce lattice of interpretations



Sequence -> Sequence: OCR

Step2: score each hypothesis



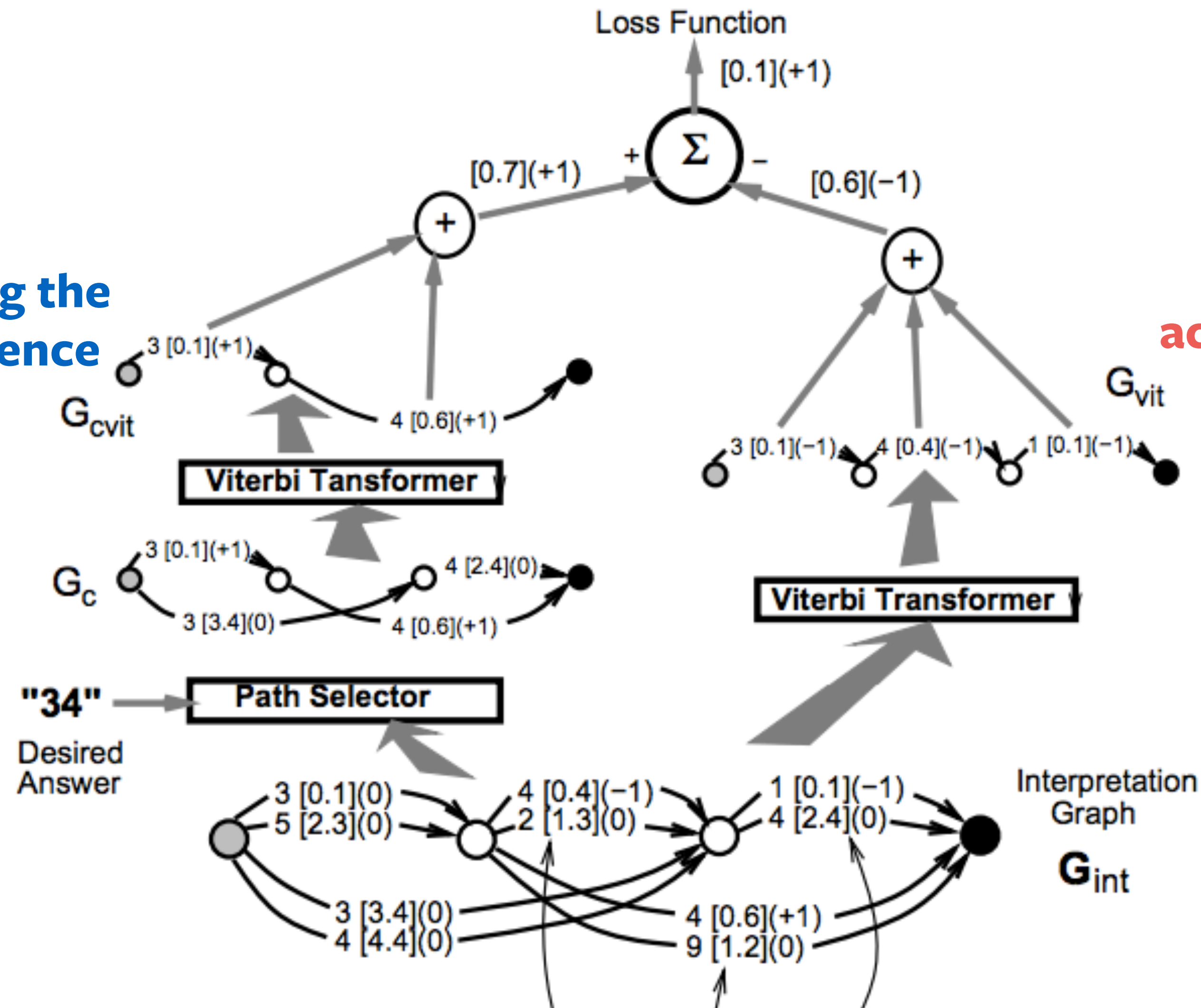
Gradient-based learning applied to document recognition, LeCun et al. IEEE 1998

Sequence -> Sequence: OCR

Step3: compute loss and gradients

Find all paths yielding the correct output sequence

Find the best path according to the model



Sequence -> Sequence: OCR

Conclusions:

- problem may have latent variables (segmentation), over which one can minimize or marginalize over.
- structure prediction is well expressed in terms of weighted lattices, and bprop still applies (GTN).
- loss functions and EBMs can straightforwardly be extended to handle sequences. This is one of the best examples of training at the sequence level.
- search over best hypothesis of the system can be expensive; marginalization can be intractable. It's problem and model dependent.

Conclusions

- sequences can appear at the input, output, or both.
- structured outputs are the most difficult case, overall when there may be several plausible predictions for the same input (e.g., MT, image captioning).
- sometimes, we do not need to bother taking into account the sequential aspect of the data, if the prediction task is well correlated to variables present in static input.
- it's possible to learn to generate sequences, to search in the space of sequences, and to still train by back-propagation as in GTNs.
- ultimately, there is no general model/loss that work in all cases. They should be designed for the task at hand.
- there are lots of demos and code available to reproduce these examples. See pytorch and torch tutorials, for instance.

